Reputation Management Service for Peer-to-Peer Enterprise Architectures

M. Amoretti, M. Bisi, M.C. Laghi, F. Zanichelli, and G. Conte

Distributed Systems Group - Dip. di Ingegneria dell'Informazione
Università degli Studi di Parma, Italy
{amoretti,laghi,zanichelli,conte}@ce.unipr.it,
matteo.bisi@sinfopragma.it

Abstract. The high potential of P2P infrastructures for enterprise services and applications both on the intranet (*e.g.* project workgroups) and on the Internet (*e.g.* B2B exchange) can be fully achieved provided that robust trust and security management systems are made available.

This paper presents the reputation system we have devised for SP2A [1], a P2P framework which supports secure role-based peergroups and service interactions. Our solution includes decentralized trust and security management able to cope with several threats. The underlying analytical model is introduced and discussed, together with a simulation-based evaluation of the robustness against malicious negative feedbacks.

Keywords: Service Oriented Architectures, Peer-to-Peer, Security, Reputation Management.

1 Introduction

Ubiquitous access to networks is deeply changing the ways enterprises organize and perform their business both internally and externally. Intranets and the global Internet allow for seamless and almost instantaneous information and knowledge sharing within organizations thus enabling more efficient processes and activities and giving rise to novel forms of interaction and supporting applications.

Peer-to-peer (P2P) technologies have gained world-wide popularity due to the success of file-sharing applications (and the predictable reactions of copyright holders) and their decentralized nature appears promising also to the purposes and applications of enterprises. P2P-based instant messaging and file-sharing can be effectively exploited on an intranet to support for example projects workgroups, distributed offices and distributions chains for documents and archives. Internet-enabled inter-firm collaboration can benefit from a P2P approach as well. Business-to-business exchanges are becoming increasingly important and many B2B communities organize themselves to be more competitive in specialized industry sectors by increasing the efficiency of their procurement and supply chains. By leveraging upon P2P technologies, the common tasks of searching for new business partners and exchanging transaction information (*e.g.* quotations) can be improved in terms of instant information, control over shared data (mantained at each P2P node) and reduced infrastructure costs.

J. Filipe and M.S. Obaidat (Eds.): ICETE 2006, CCIS 9, pp. 52–63, 2008.

The vision of unmediated, instantaneous trading as well as more realistic P2P-based B2B communities can be approached only if enterprise-level solutions are made available to cope with the fundamental trust and security issues. *Identity trust*, namely the belief that an entity is what it claims to be, can be assessed by means of an authentication scheme such as X.509 digital identity certificates. *Provision trust*, that is the relying party's trust in a service or resource provider, appears more critical as users require protection from malicious or unreliable service providers. Unlike B2B exchanges based on centralized, third party UDDI directories which offer trustworthy data of potential trading partners (*i.e.* service providers), P2P decentralized interaction lends itself to trust and reputation systems mainly based on first hand experience and second-hand referrals. This information can be combined by a peer into an overall rating or reputation value for a service provider and should influence further interaction with it.

In this paper we present the reputation system we have devised for SP2A [1], a P2P framework which supports secure role-based peergroups and service interactions. Our system includes decentralized trust and security management able to cope with several threats, starting from *impersonification*, which refers to the threat caused by a malicious peer posing as another in order to misuse that peer's privileges and reputation. Digital signatures and message authentication are typical solutions for this kind of attack. As malicious peers can engage in *fraudolent actions*, such as advertising false resources or services and not fulfilling commitments, a consistent reputation management system has been introduced in our P2P framework which also forbids *trust misrepresentation* attempts. In a peer-to-peer system, the most difficult threat to discover and neutralize is *collusion*, which refers to a group of malicious peers working in concert to actively subvert the system. To face this danger, the default policy provided by our security framework is *role-based group membership based on secure credentials (SC policy)*. Based on this strategy, a group of peer can filter what actions its members can perform. This paper focuses on reputation management, which was left as open issue in our previous work [2].

Next section 2 outlines the issues of reputation management systems and the choices available for centralized and decentralized implementation. The analytical model underlying our reputation management is described in section 3. An emulation scenario is then presented, first describing a four roles configuration example (section 4) and then discussing the obtained results (section 5). Section 6 reports on some relevant work in the area of reputations systems for P2P systems. Finally, a few conclusive remarks and an indication of further work conclude the paper.

2 Reputation Management in Role-Based Peergroups

A role-based peergroup can achieve stability only if each participant (which is supposed to be authenticated and authorized) bases its actions on previous experience and/or recommendations, *i.e.* which define the reputation of the other participants. Reputation and trust are orthogonal concepts, which require, in a peer-to-peer context, complex management mechanisms such as node identification and digitally signed certificates exchange [3].

Our model considers both *peer reputation* and *service reputation*. A peer can provide more than one service, each of them with its own reputation which contributes to the overall reputation of the peer. Peer reputations may have non-zero initial value, fixed by a trusted third party. Based on peer's behaviour, the reputation value changes with time, and represents the trustworthiness of peers on the basis of their transaction with other nodes. Each peer, at the end of an interaction with another member of the group, can provide its feedback about each consumed service; the feedback is used to update the reputation of the provider peer.

Two issues arise: where are to be stored the reputation information, and how to guarantee their integrity. Several solutions can be adopted:

1. a stable and recognized peer stores and manages the reputation information of all group members (*centralized* solution);
2. each peer stores its experience against other peers, and when others ask for reputation information of a particular peer, it provides an answer based on its stored information (*local* solution);
3. the reputation storage is partitioned into several small parts, which are stored in all peers; that is, every peer equally manages some part of the whole reputation information (*global* solution);
4. only stable, recognized and highly-reputed peers are reputation collectors (*mediated* solution).

It can be easily seen that not all these solutions are equally scalable, efficient and robust.

Solution 1 is easy to implement as a Centralized Reputation Management Service (CRMS), but it does not scale well, *i.e.* it could work only for small peergroups. Solutions 2, 3 and 4 require a Distributed Reputation Management Service (DRMS).

The local solution is slightly efficient and lacks robustness. If a peer wants more objective reputation about another peer, it should ask as many peers as necessary, thus generating a lot of messages in the peer-to-peer network. Moreover, if the reputation information is concentrated in few very active customer peers, the reputation system becomes broken when these are not online.

The global solution is very attractive, in particular if the reputation management system is implemented as a Distributed Hash Table (DHT). In this case, the peer which is responsible for a specific reputation information is determined with a hash function within $O(1)$ time, and its location is found within $O(\log N)$ time.

The mediated solution should be appropriated for unstructured networks, with few highly connected and stable nodes, *e.g.* scale-free topologies [4].

3 Analytical Model

In this section we illustrate an analytical model which defines the fundamental parameters which are involved in the evolution of the reputation, for each role which can be taken in a peergroup based on our SC policy.

The reputation Rep of a peer is the difference between the number of positive feedbacks and the number of negative feedbacks. Feedbacks take values $+1$ and -1 with probability p and $q = 1 - p$ respectively. Thus, the sequence of feedbacks that a peer receives can be considered as a generalized random walk, a stochastic process which becomes nearly normal after long time.

When a peer joins the secure peergroup, and each time it is promoted or demoted, it receives an initial reputation value which is stored by the reputatation management service. We consider the following parameters:

- *total votes* $n = n_+ + n_-$, which represents the sum of received feedbacks;
- *good ratio* $R = \frac{n_+}{n_+ + n_-}$, which is the number of positive feedbacks, versus n; R is a fundamental parameter for the analytical model, because its instantaneous value allows to define the *dependability degree* of the peer.

The domain of R and n can be easily obtained from their definitions:

$$0 \leq R \leq 1, n \geq 0$$

Depending on values of R and n, we can consider four different conditions for each role a peer can take, which are listed below.

- $n \geq n_{th}$, where $n_{th} \geq 0$ is the *confidence threshold*, evaluated on all received feedbacks. The reason of considering such a threshold is that the dependability of the reputation value of a peer depends on the total number of performed transactions (the more they are, the more the value is dependable).
- $n < n_{th}$ means that the peer has recently joined the group, thus the reputation value must be weighted to consider a potentially less dependability.
- $R \geq R_{th}$, where $0 \leq R_{th} \leq 1$ is the *trust threshold*. In this case, the peer can be trusted.
- $R < R_{th}$, on the other side, means that the peer cannot be trusted and should be demoted.

From the definitions of n and R, we obtain

$$Rep = n_+ - n_- = (2R - 1)n \tag{1}$$

from which we can derive the *role preservation condition*

$$Rep \geq (2R_{th} - 1)n \tag{2}$$

Suppose that R, in the steady state, is characterized by little variations around an average value. A sudden increase or decrease of R may be considered as a suspect event. If an abrupt increase or decrease of R lasts for a significant time interval, possibily leading R to its upper or lower threshold (R^u_{th} or R^l_{th}, respectively), an attack may be in progress.

In order to avoid misleading demotions or promotions, a simple but effective solution is to introduce two *braking windows*, one for the upper threshold and one for the lower threshold, within which R is increased by R_b, defined as a non-linear function of R with the following properties:

- R_b is positive and between R^l_{th} and R^l_w
- R_b is zero between R^l_w and R^u_w
- R_b is negative between R^u_w and R^u_{th}

Figure 1) illustrates an example of a particular $R_b(R)$ function.

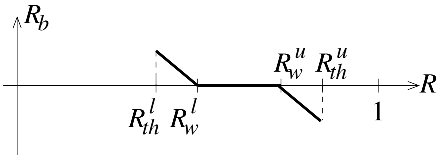

Fig. 1. An example of $R_b(R)$ with decreasing shape and braking windows of the same size

The mechanism of the braking windows is general, although the window size should be different for each role, namely decreasing with the importance of the peer.

It is possible to compute from previous parameters the rate r_{bad} with which one or more (possibly cooperating) malicious peers can provide negative feedbacks without affecting the peer's role. Above this rate value, the peer is soon demoted for insufficient good ratio, according to the rules which have been illustrated above.

We first consider the case with no braking windows. In the time unit Δt, the average number of received feedbacks is defined as

$$\Delta n = \Delta n_+ + \Delta n_- + \Delta n_{bad}$$

where Δn_+ and Δn_- are honest feedbacks, while Δn_{bad} represents deceptive negative feedbacks, sent by malicious peers. For sake of simplicity we suppose that no deceptive positive feedbacks are received, and $\Delta n_+ \geq \Delta n_-$. The reputation decreases if

$$\Delta n_{bad} > \Delta n_+ - \Delta n_- \tag{3}$$

i.e.

$$r_{bad} = \frac{\Delta n_{bad}}{\Delta n} > 2R - 1 \tag{4}$$

At the trust threshold, this means $r_{bad} > 2R_{th} - 1$.

If there are the braking windows, the adjusted good ratio is

$$R' = R + R_b(R)$$

thus the value of r_{bad} which leads to peer demotion becomes

$$r_{bad} > 2[R + R_b(R)] - 1 \tag{5}$$

4 Four-Role Configuration Example

We now apply the analytical model to an example of a four-role reputation policy of a service-sharing system. For each role we set the initial values for the parameters we illustrated in section 3, considering for simplicity only the lower threshold and the related braking window. In details, the list of roles is:

- admin - the peer is highly-reputed, and trusted by the group founder, or it is the group founder itself; the actions it is allowed to perform are: service sharing/discovery, group monitoring, voting for changing member ranks, store reputation information (if the mediated solution is adopted);
- newbie - the peer is a new member; it only can search for an admin peer, to ask for a promotion;
- searcher - the peer is allowed to search for services and to interact with them;
- publisher - the peer can search for services but also publish its own services in the peergroup.

Each **admin** peer has the following configuration:

$$\begin{cases} n_{+,init} = 50 \\ n_{-,init} = 10 \\ n_{init} = n_{+,init} + n_{-,init} = 60 \\ R_{init} = \dfrac{n_{+,init}}{n_{+,init} + n_{-,init}} = 0.8 \\ Rep_{init} = n_{+,init} - n_{-,init} = 40 \\ R_{th} = 0.6 \\ R_w = 0.8 \\ R_b(R_{th}) = 0.075 \end{cases}$$

For example, we assume $\Delta n_+ = 9.6$ and $\Delta n_- = 2.4$ in the average, *i.e.* $\Delta n = 12$ feedbacks per time unit. Without braking window, the number of deceptive negative feedbacks per time unit must be $\Delta n_{bad} < 3$, since $r_{bad} < 2R_{th} - 1 = 0.2$. With the braking window, the number of deceptive negative feedbacks per time unit must be $\Delta n_{bad} < 6,46$, since $r_{bad} < 2(R_{th} + R_b(R_{th})) - 1 = 0.35$. Thus the braking window makes the system more robust.

Each **publisher** peer has the following configuration:

$$\begin{cases} n_{+,init} = 35 \\ n_{-,init} = 10 \\ n_{init} = n_{+,init} + n_{-,init} = 45 \\ R_{init} = \dfrac{n_{+,init}}{n_{+,init} + n_{-,init}} = 0.\overline{7} \\ Rep_{init} = n_{+,init} - n_{-,init} = 25 \\ R_{th} = 0.6 \\ R_w = 0.75 \\ R_b(R_{th}) = 0.04 \end{cases}$$

Compared with previous case, the braking window for a publisher is smaller: 0.15 versus 0.2. With this window, the maximum tolerable rate of deceptive negative feedbacks is $r_{bad} = 28\%$.

Each **searcher** peer has the following configuration:

$$\begin{cases} n_{+,init} = 25 \\ n_{-,init} = 10 \\ n_{init} = n_{+,init} + n_{-,init} = 35 \\ R_{init} = \dfrac{n_{+,init}}{n_{+,init} + n_{-,init}} = 0.714 \\ Rep_{init} = n_{+,init} - n_{-,init} = 15 \\ R_{th} = 0.6 \\ R_w = 0.7 \\ R_b(R_{th}) = 0.025 \end{cases}$$

For a searcher, whose braking window is 0.1 large, the maximum tolerable rate of deceptive negative feedbacks is $r_{bad} = 25\%$.

Finally, each **newbie** peer has the following configuration:

$$\begin{cases} n_{+,init} = 15 \\ n_{-,init} = 10 \\ n_{init} = n_{+,init} + n_{-,init} = 25 \\ R_{init} = \dfrac{n_{+,init}}{n_{+,init} + n_{-,init}} = 0.6 \\ Rep_{init} = n_{+,init} - n_{-,init} = 5 \\ R_{th} = 0.6 \\ R_w = 0.65 \\ R_b(R_{th}) = 0.005 \end{cases}$$

Thus a newbie, which is characterized by the smallest braking window (0.05), can tolerate at most $r_{bad} = 21\%$.

5 Emulation Results

SP2A is an abstract framework but also a Java middleware for the development and deployment of service-oriented peer-to-peer architectures. Using its simple API, we realized a centralized reputation management service able to emulate the interaction of that service with an hypothetical network of peers which provide positive and negative feedbacks. We emphasize that our purpose was not to evaluate reputation retrieval and maintainance performance, obviously radically different in the cases of centralized and distributed services. The deployed testbed allowed us to verify the correctness of the proposed analytical model as well as the tuning of the parameter values for the four-role secure group configuration.

The reputation management service maintains a reputation table, and randomly assigns feedbacks to peers, with $\Delta n = 12$ as assumed in section 4. All emulations started with each peer having $R = R_{init}$, and lasted the time necessary to observe significant results (we set $\Delta t = 1$ minute). We tracked the evolution of the reputation value Rep for each peer, and we computed the average behaviour for each role. We initially emulated a peergroup of righteous peers, in which positive and negative feedbacks per unit time are distributed with $R_{avg} = R_{init}$. Then, we performed several emulations of a system in which some peers provided deceptive negative feedbacks with increasing rate. For each role, we found the maximum tolerable rate of malicious negative feedbacks, over which the target peer is demoted, or banned from the peergroup if its role is newbie.

In general, to know if the role preservation condition $Rep \geq (2R_{th} - 1)n$ will be fulfilled, in a stable condition with fixed r_{bad} and R_{avg}, we need to compare the average slope of the current reputation curve, with the average slope of the minimum reputation curve representing $Rep(R_{th})$ (using, for example, the least squares method). There are two possible situations:

- $\frac{d}{dt}Rep \geq \frac{d}{dt}Rep_{th}$: the curves diverge, *i.e.* the reputation of the peer increases more quickly than the minimum reputation, under which the peer is demoted;
- $\frac{d}{dt}Rep < \frac{d}{dt}Rep_{th}$: the curves converge, *i.e.* in a non-infinite time the peer will be demoted.

Also note that both curves depend on r_{bad}, because $\Delta n = \Delta n_{+} + \Delta n_{-} + \Delta n_{bad}$.

Starting from $R = R_{init}$, the good ratio decreases if the number of negative feedbacks per time unit is higher than the number of positive feedbacks. In particular, this eventuality can arise if $r_{bad} > 0$. In our emulations, for each role we set a braking window (according to the parameters illustrated in section 4), which enters the game when $R < R_w$, and contributes to maintain $\frac{d}{dt}Rep \geq \frac{d}{dt}Rep_{th}$.

Figure 2 illustrates the average evolution of an admin peer's reputation over the emulation time interval. If no malicious peers provide deceptive negative feedbacks, the measured reputation of the target peer is represented by the fat continuous curve. Comparing this curve with the graph of the reputation which we obtain if $R = R_{th}$ (the thin continuous curve in the figure), we can observe that they diverge, thus we expect that the peer will not be demoted unless $r_{bad} = 0$. In the same figure, dotted curves refer to the case of $r_{bad} = 20\%$; they still diverge. Finally, dashed curves show the limit over which the role preservation condition is not fulfilled, *i.e.* $r_{bad} = 35\%$, the same we computed with the analytical model.

The most interesting emulation results for the publisher role are illustrated in figure 3, which compares the case of no malicious peers (continuous lines) with the case of deceptive negative feedbacks with $r_{bad} = 29.4\%$ rate (dashed line). We can observe that, in the latter case, the average slopes show that the curves eventually converge. We measured $max\{r_{bad}\} = 28\%$, over which the publisher is demoted in a non-infinite time. Also this result is compatible with the analytical model.

Figures 4 and 5 illustrates, respectively, emulation results for the searcher and the newbie roles. We measured a maximum malicious feedbacks rate $r_{bad} = 25\%$, for a searcher peer. The figure illustrates what happens when this rate is overthrown, *i.e.* the reputation curves (average and minimum) converge. For a newbie peer, the measured maximum rate of deceptive negative feedbacks is $r_{bad} = 21\%$. The figure illustrates a less dangerous situation. Both these emulations gave satisfactory results, which respect the numerical constraints obtained with the analytical model.

All results are summarized in table 1. The first and second columns report the analytical results, respectively for a reputation management service without and with braking windows. The third column reports the emulation results, which refer to a reputation management service with braking window and target peer with righteous behaviour,

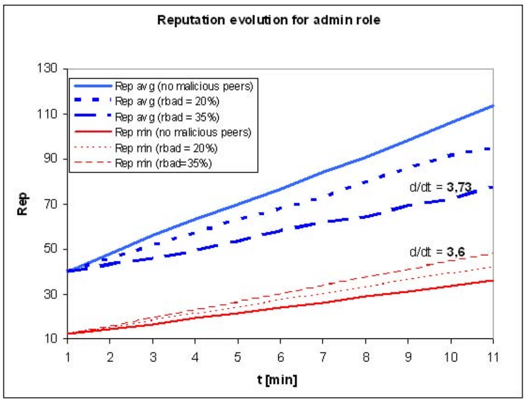

Fig. 2. Average and minimum reputation dynamics, for an admin peer, for different rates of malicious negative feedbacks

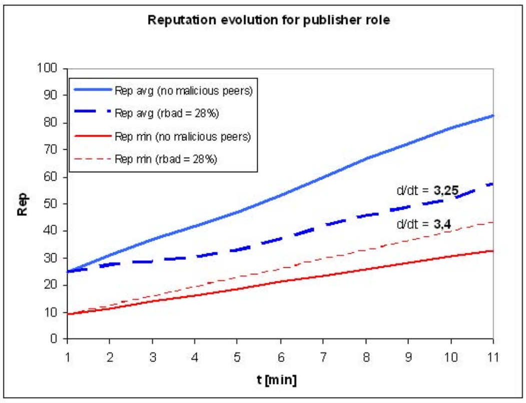

Fig. 3. Average and minimum reputation dynamics, for a publisher peer, for different rates of malicious negative feedbacks

i.e. a peer which would maintain the initially assigned good ratio R_{init} in absence of malicious negative feedbacks.

6 Related Work

There have been several studies about managing reputation in P2P networks, most of them related to content sharing, which is currently the killer application for these architectures. A reputation management system in DHT-based structured P2P networks is proposed in [5]; this model uses file reputation information as well as peer reputation information, and the system uses a global storage for reputation information, that is available when evaluator is not on-line. The reputation information consists, as in our model, of two values representing the number of positive and negative feedbacks.

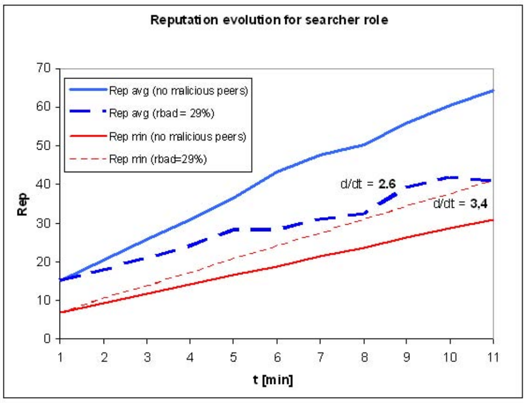

Fig. 4. Average and minimum reputation dynamics, for a searcher peer, for different rates of malicious negative feedbacks

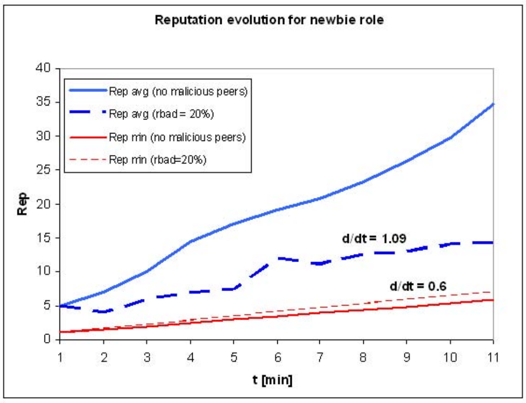

Fig. 5. Average and minimum reputation dynamics, for a newbie peer, for different rates of malicious negative feedbacks

In [6] a reputation management system for partially-decentralized P2P systems is described, in which the reputation information is managed by supernodes. The authors assume that the supernodes are selected from a set of trusted peers, and they share a secret key used to digitally sign the reputation data. Good reputation is obtained by having consistent good behaviour through several transactions. The proposed scheme is based on four values associated to each peer and stored at the supernode level; two of them are used to provide an idea about the satisfaction of users, and the others express the amount of uploads provided by the peer. The reputation information is updated according with the peer transactions of upload and download. In a posterior work [7] the same authors propose an algorithm to detect malicious peers which are sending inauthentic files or are lying in their feedbacks. They distinguish righteous peers from those which share

Table 1. Maximum tolerable malicious feedbacks rate r_{bad}: analytical results without and with braking window, and simulated results

Role	th_{norec}	th_{rec}	sim_{rec}
A	20%	35%	35%
P	20%	28%	28%
S	20%	25%	25%
N	20%	21%	21%

inauthentic files and provide false feedbacks about other peers. The model introduces the concept of suspicious transaction, that is a transaction whose appreciation depends on the reputation of the sender and the concept of credibility behaviour, as an indicator of the liar behaviour of peers. These schemes are able to detect malicious peers and isolate them from the system, but do not consider the tolerable rate of malicious negative feedbacks, and suppose that supernodes are always trustworthy.

A distributed method to compute global trust values, based on power iteration, is illustrated in [8]. The reputation system aggregates local trust values of all users, by means of an approach based on transitive trust: a peer will have a high opinion of those peers which have provided authentic files and it is likely to trust the opinions of those peers, since peers which are honest about the files they provide are also likely to be honest in reporting their local trust values. The scheme is reactive, *i.e.* it requires reputations to be computed on-demand, through the cooperation of a large number of peers. This introduces additional latency and requires a lot of time to collect statistics and compute the global rating.

The protocol proposed in [9] aims to distinguish malicious responses from benign ones, by using the reputation of the peers which provide them. The protocol relies on the P2P infrastructure to obtain the necessary reputation information when it is not locally available at the querying peer. The outcomes of past transactions are stored in trust vectors; every peer maintains a trust vector for every other peer it has dealt with in the past. The trust query process is similar to the file query process except that the subject of the query is a peer about whom trust information is inquired. The responses are sorted and weighted by the credibility rating of the responder, derived from the credibility vectors maintained by the local peer, which are similar to the trust vectors.

In [10], the use of a scheme named ROCQ (Reputation, Opinion, Credibility and Quality) in a collaborative content-distribution system is analyzed. ROCQ computes global reputation values for peers on the basis of first-hand opinions of transactions provided by participants. Global reputation values are stored in a decentralized fashion using multiple score managers for each individual peer. The final average reputation value is formed by two aggregations, first at the score managers and second at the requesting peer.

All these works consider the situation in which a peer with a bad reputation is simply isolated from the system, while the analytical model we are proposing describes different roles for peers, associated with different actions. So a peer with a suspect malicious behaviour can be first demoted, and eventually isolated from the system.

7 Conclusions

In this work we have illustrated the analytical model of a reputation management service for role-based peergroups. The model defines some parameters and indicators, such as the maximum tolerable rate of malicious negative feedbacks. We applied the reputation model to an example of four-role security policy, giving a parameter set for each role, and computing the theoretical values for the main indicators. These results have been confirmed by those we obtained from several emulations exploiting a centralized reputation management service.

Further work will follow two main directions. In order to improve the analytical model, we need to compare the effectiveness of different functions for realizing the braking windows. Once the analytical model is verified and all of its parameters are tuned, we will investigate an efficient and robust distributed solution for reputation storage and retrieval by means of simulations and prototypes.

Acknowledgements

This work has been partially supported by the "STIL" project of Regione Emilia-Romagna, and by the "WEB-MINDS" FIRB project of the National Research Ministry.

References

1. Amoretti, M., Zanichelli, F., Conte, G.: SP2A: a Service-oriented Framework for P2P-based Grids. In: 3rd International Workshop on Middleware for Grid Computing, Co-located with Middleware (2005)
2. Amoretti, M., Bisi, M., Zanichelli, F., Conte, G.: Introducing Secure Peergroups in SP2A. In: HOTP2P 2005, San Diego, California, USA (2005)
3. Ye, S., Makedon, F., Ford, J.: Collaborative Automated Trust Negotiation in Peer-to-Peer Systems. In: Fourth International Conference on Peer-to-Peer Computing (P2P 2004) (2004)
4. Barabási, A.-L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)
5. Lee, S.Y., Kwon, O.H., Kim, J., Hong, S.J.: A Reputation Management System in Structured peer-to-peer Networks. In: The 14th International Workshops on Enabling Technologies:Infrastruture for Collaborative Enterprise (2005)
6. Mekouar, L., Iraqui, Y., Boutaba, R.: A Reputation Management and Selection Advisor Schemes for peer-to-peer Systems. In: The 15th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (2004)
7. Mekouar, L., Iraqui, Y., Boutaba, R.: Detecting Malicious Peers in a Reputation-based peer-to-peer System (2004)
8. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust Algorithm for Reputation Management in peer-to-peer Networks. In: The 12th International World Wide Web Conference (2003)
9. Selcuk, A.A., Uzun, E., Pariente, M.R.: A Reputation-Based Trust Management System for peer-to-peer peer-to-peer Networks. In: IEEE International Symposium on Cluster Computing and the Grid (2004)
10. Garg, A., Battiti, R., Cascella, R.: Reputation Management: Experiments on the Robustness of rocq (2005)